Executorch XNNPACK Quantization

Export

1. Downloads Checkpoint & Tokenizer

```
wget
"https://huggingface.co/karpathy/tinyllamas/resolve/main/stories110M.pt"
wget
"https://raw.githubusercontent.com/karpathy/llama2.c/master/tokenizer.model"
```

2. Edit params.json

```
echo '{"dim": 768, "multiple_of": 32, "n_heads": 12, "n_layers": 12,
"norm_eps": 1e-05, "vocab_size": 32000}' > params.json
```

3. XNNPACK Export Command

```
python -m examples.models.llama.export_llama -kv -qmode 8da4w -X --
group_size 128 -d fp32 -c stories110M.pt -p params.json
```

4. Convert Tokenizer

```
python -m extenstion.llm.tokenizer.tokenizer -t tokenizer.model -o
tokenizer.bin
```

Android Setup

Android Environments Setup

```
export ANDROID_NDK=/opt/android-sdk/ndk/26.3.11579264
export ANDROID_NDK_ROOT=/opt/android-sdk/ndk/26.3.11579264
export ANDROID_HOME=/opt/android-sdk
```

Build executorch library for Android

```
cmake -DCMAKE_TOOLCHAIN_FILE=$ANDROID_NDK/build/cmake/android.toolchain.cmake \
   -DANDROID_ABI=arm64-v8a \
   -DANDROID_PLATFORM=android-23 \
   -DCMAKE_INSTALL_PREFIX=cmake-out-android \
   -DCMAKE_BUILD_TYPE=Release \
   -DEXECUTORCH_BUILD_EXTENSION_DATA_LOADER=ON \
   -DEXECUTORCH_BUILD_EXTENSION_MODULE=ON \
   -DEXECUTORCH_BUILD_EXTENSION_TENSOR=ON \
   -DEXECUTORCH_ENABLE_LOGGING=1 \
   -DPYTHON_EXECUTABLE=python \
   -DEXECUTORCH_BUILD_XNNPACK=ON \
   -DEXECUTORCH_BUILD_KERNELS_OPTIMIZED=ON \
   -DEXECUTORCH_BUILD_KERNELS_QUANTIZED=ON \
   -DEXECUTORCH_BUILD_KERNELS_CUSTOM=ON \
   -Bcmake-out-android .

cmake --build cmake-out-android -j16 --target install --config Release
```

Build llama runner for Android

```
cmake  -DCMAKE_TOOLCHAIN_FILE=$ANDROID_NDK/build/cmake/android.toolchain.cmake \
   -DANDROID_ABI=arm64-v8a \
   -DANDROID_PLATFORM=android-23 \
   -DCMAKE_INSTALL_PREFIX=cmake-out-android \
   -DCMAKE_BUILD_TYPE=Release \
   -DPYTHON_EXECUTABLE=python \
   -DEXECUTORCH_BUILD_XNNPACK=ON \
   -DEXECUTORCH_BUILD_KERNELS_OPTIMIZED=ON \
   -DEXECUTORCH_BUILD_KERNELS_QUANTIZED=ON \
   -DEXECUTORCH_BUILD_KERNELS_CUSTOM=ON \
   -Bcmake-out-android/examples/models/llama \
   examples/models/llama

cmake --build cmake-out-android/examples/models/llama -j16 --config Release
```

Build AAR Library

```
bash examples/demo-apps/android/LlamaDemo/download_prebuilt_lib.sh
```

And then, **executorch.aar** file will be generated in a newly created folder in **examples/demo-apps/android/LlamaDemo/app/libs** directory.

Upload Exported Model, Tokenizer and llama runner binary files

```

```

```
adb shell mkdir -p /data/local/tmp/llama/
adb push <llama2.pte> /data/local/tmp/llama/
adb push tokenizer.bin /data/local/tmp/llama/
adb push cmake-out-android/examples/models/llama/llama_main /data/local/tmp/llama/
```

Now, You can try to run the model on your device

```
adb shell
cd /data/local/tmp/llama
chmod 777 llama_main
./llama_main --model_path llama2.pte --tokenizer_path tokenizer.bin --prompt Once
upon a time,
```